REMARKS

Entry of this Amendment is proper because it does not raise any new issues requiring further search by the Examiner, narrows the issues on appeal, and is believed to place the present application in condition for immediate allowance.

Claims 1-3 and 5-40 are all the claims presently pending in the application.

While Applicant believes that claim 1 is patentable over the cited references, either alone or in combination, to define more clearly the features of the claimed invention and to speed prosecution, Applicant amends claim 1 to incorporate all of the features of dependent claim 4. Applicant respectfully notes that no additional search should be necessary since original claim 4 presumably was fully examined and considered in the previous Office Action. Claim 4 correspondingly is canceled without prejudice or disclaimer.

It is noted that the claim amendments are made only for more particularly pointing out the invention, and not for distinguishing the invention over the prior art, narrowing the claims or for any statutory requirements of patentability. Further, Applicants specifically state that no amendment to any claim herein should be construed as a disclaimer of any interest in or right to an equivalent of any element or feature of the amended claim.

Claims 1-26 and 28-40 stand rejected under 35 U.S.C. § 103(a) as being obvious from Fulton, III et al. (U.S. Patent No. 5,715,386; hereinafter "Fulton") in view of Garg ("A Methodology for Detection and Estimation of Software Aging", published November 1998. Claim 27 stands rejected under 35 U.S.C. § 103(a) as being obvious over Fulton in view of Garg, and further in view of Murphy, et al. (U.S. Publication No. 2003-015084 A1, filed January 10, 2000).

These rejections are respectfully traversed in the following discussion.

I. THE CLAIMED INVENTION

In an illustrative, non-limiting aspect of the present application, as defined by independent claim 1, a method of reducing a time for a computer system to recover from a degradation of performance in a hardware or a software in at least one first node of the computer system, includes monitoring a state of the at least one first node, and based on the monitoring, transferring a state of the at least one first node to a second node prior to the degradation in performance of the hardware or the software of the at least one first node. The method further includes proactively invoking a state migration functionality to reduce said recovery time.

In another exemplary aspect of the invention, as defined by independent claim 19, a method of reducing a lack of performance in a computer system having at least one primary node and a secondary node, includes determining whether a failure or lack of performance is imminent, based on said determining, commanding a secondary node to start an application if it is not already running, and to begin reading a state and redo log from a memory coupled to said primary node and said secondary node, commanding the secondary node to apply the redo log to its state, commanding the primary node to begin mirroring its dynamic state updates to the secondary node as they occur, such that the secondary node's state is brought completely up to date with said primary node, judging whether the primary node has failed, and based on said judging, making the secondary node become the primary node.

Independent claims 26-28 and 34-40 recite somewhat similar features as independent claims 1 and 19.

In conventional devices, when a computer system suffers an unplanned failure, a certain amount of time is required to recover from the failure. This outage duration is undesirable.

For example, if the computer is a single node, stand-alone computer system, it must reboot and restart its application. If the computer is part of a multi-node high availability cluster architecture, it must failover (i.e., transfer) the application to another node in the cluster. During this recovery time, after either rebooting or failing-over the application to another node in a cluster environment, the recovering system must reload a stale copy of its state from disk, load a transaction redo log from disk, and attempt to reconstruct an up-to-date copy of that state by replaying that transaction redo log against the stale state (e.g., see specification at page 2, lines 2-11).

The claimed invention, on the other hand, provides an exemplary method (and system) for proactively reducing the outage duration by using the predicted outages to proactively trigger and manage existing failure recovery functionality (e.g., see specification at page 1, lines 13-16).

Particularly, the claimed invention exploits the ability to predict software outages or hardware failures to proactively migrate the state needed to quickly recover from an imminent outage onto another computer system (e.g., such as another node in a cluster environment) or a persistent storage medium (e.g., such as a hard disk) (e.g., see specification at page 5, lines 10-18). According to the present invention, in an exemplary, non-limiting aspect, the system that was about to fail (e.g., the "failing computer") could "prime" another computer (e.g., the "failover target") by directing it to read from disk the stale state and the redo log, and then direct the failover target computer to begin applying the redo log to the stale state. Moreover, upon discovery of its imminent demise, the failing computer could mirror all transactions to the failover target, bringing the failover target's state even more up-to-date. When the primary computer does finally fail, the failover target would have an up-to-date copy of the state, the lengthy reloading of state and redo log from disk would be avoided, the outage would be

shortened, and the system availability would be improved (e.g., see specification at page 6, lines 7-12).

Thus, the present invention reduces outage duration by using the ability to predict outages to proactively trigger and execute functionality whose effect is to reduce that outage's duration. Further, in the case of a single node, the amount of time required to reconstruct the node's state after the outage has occurred can be reduced significantly (e.g., see specification at page 6, lines 7-12).

II. THE PRIOR ART REJECTIONS

A. Claims 1-26 and 28-40 stand rejected under 35 U.S.C. § 103(a) as being obvious from Fulton in view of Garg.

Applicant incorporates herein by reference in their entirety the traversal arguments set forth in the Amendment under 37 C.F.R. § 1.111 filed on September 30, 2004, for the Examiner's convenience.

In the Response to Arguments set forth in the present Office Action, the Examiner alleges that "*Fulton does, in fact, disclose a method that reduces time spent recovering from performance degradation. Through the use of the forced recovery of the rejuvenation actions, the number of failures are reduced and any time spent in recovery from these failures is also reduced, resulting in less downtime for the system implemented by Fulton (Fulton, col. 5, lines 55-61)*" (see Office Action at page 2, lines 3-10; emphasis Applicant's). The Examiner also alleges that "*Fulton does in fact reduce the outage duration overall by avoiding the longer outages due to unplanned failures (Fulton, col. 5, lines 55-61(sic))*" (see Office Action at page 2, lines 10-13; emphasis Applicant's).

The Examiner further alleges that "*a proper combination of Fulton and Garg would result in a system that uses the monitoring of Garg to detect the best time to implement the rejuvenation of Fulton, which would reduce outage by making more efficient use of each outage period (Fulton, col. 5, lines 55-61)*" (see Office Action at page 2, lines 14-19; emphasis Applicant's). The Examiner also alleges that "*Fulton fully discloses the reduction of time to recovery as stated above, and that the use of Murphy is not necessary in this aspect*" (see Office Action at page 3, lines 1-4).

However, Applicant respectfully disagrees with the Examiner's stated positions for several reasons, and therefore, respectfully traverses these rejections.

Applicant submits that the cited portion of the disclosure of Fulton (at column 5, lines 55-61) clearly does not disclose or suggest the features of the claimed invention, as defined by independent claims 1, 19, 26-28, and 34-40.

Instead, Fulton discloses that "*[i]f rejuvenation is performed during the most idle time of the application, then ... the total expected downtime cost in the application with rejuvenation, as computed in Equation 6, would be lower than that without rejuvenation as computed in Equation 4*" (see Fulton at column 5, lines 55-60). That is, Fulton merely discloses that performing the rejuvenation during a most idle time takes less time than performing rejuvenation during a non-idle time.

However, Applicant respectfully submits that performing rejuvenation during the most idle time of the application so that rejuvenation time is less than when the rejuvenation is performed during other times (i.e., non-idle times) clearly is different than the claimed invention, which proactively reduces the outage duration by using the predicted outages to proactively

trigger and manage existing failure recovery functionality (e.g., see specification at page 1, lines 13-16).

That is, unlike Fulton, the claimed invention exploits the ability to predict software outages or hardware failures to proactively migrate the state needed to quickly recover from an imminent outage onto another computer system (e.g., such as another node in a cluster environment) or a persistent storage medium (e.g., such as a hard disk) (e.g., see specification at page 5, lines 10-18).

Thus, the claimed invention reduces outage duration for that outage by using the ability to predict outages to proactively trigger and execute functionality whose effect is to reduce that outage's duration (e.g., see specification at page 6, lines 7-12), not merely reducing the relative outage time or duration of one outage with respect to other outages by performing the rejuvenation of one outage during a most idle time so that it takes less time than performing other rejuvenation outages during a non-idle time (i.e., performing a rejuvenation during a most idle time takes less time relative to the time to perform a rejuvenation during a non-idle time), as described in Fulton.

Thus, when considered as a whole for what it fairly teaches to the ordinarily skilled artisan, Fulton clearly does not disclose or suggest all of the features of the claimed invention.

Indeed, Applicant notes that distilling the invention down to the "gist" or the "thrust" of the invention disregards the requirement of analyzing the subject matter "as a whole" (e.g., see M.P.E.P. § 2141.02; see also *W.L.Gore & Associates, Inc. V. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), cert. denied, 469 U.S. 851 (1984)). That is, the question is whether the claimed invention as a whole would have been obvious from the cited references (e.g., see

M.P.E.P. § 2141.02; see also *Stratoflex, Inc., v. Auroquip Corp.*, 713 F.2d 1530, 218 USPQ 871 (Fed. Cir. 1983)). The Examiner must consider each and every element of all of the claims.

Independent Claim 1:

While Applicant believes that claim 1 is patentable over the cited references, either alone or in combination, to define more clearly the features of the claimed invention and to speed prosecution, Applicant amends claim 1 to include the features of claim 4. Applicant respectfully notes that no additional search should be necessary since original claim 4 presumably was fully examined and considered in the previous Office Action. Claim 4 correspondingly is canceled without prejudice or disclaimer.

For example, independent claim 1 recites a method of reducing a time for a computer system to recover from a degradation of performance in a hardware or a software in at least one first node of said computer system, comprising:

> monitoring a state of said at least one first node;
> based on said monitoring, transferring a state of said at least one first node to a second node prior to said degradation in performance of said hardware or said software of said at least one first node; and
> proactively invoking a state migration functionality to reduce said recovery time (emphasis added).

Accordingly, the claimed invention proactively reduces the outage duration by using the predicted outages to proactively trigger and manage existing failure recovery functionality (e.g., see specification at page 1, lines 13-16).

Moreover, Applicant submits that modifying Fulton to include "*the monitoring of Garg to detect the best time to implement the rejuvenation of Fulton*" would not reduce outage time of that outage, but instead, merely provide an outage time that is less than (i.e., in comparison with) some other outage time (i.e., for some other outage).

Independent claim 19

On the other hand, with respect to independent claim 19, Applicant respectfully notes that the Examiner has not responded to Applicant's traversal position set forth in the Amendment under 37 C.F.R. § 1.111 filed on September 30, 2004, in accordance with M.P.E.P. § 707.07(f).

As mentioned above, Applicant notes that distilling the invention down to the "gist" or the "thrust" of the invention disregards the requirement of analyzing the subject matter "as a whole" (e.g., see M.P.E.P. § 2141.02; see also *W.L. Gore & Associates, Inc. V. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), cert. denied, 469 U.S. 851 (1984)). That is, the question is whether the claimed invention as a whole would have been obvious from the cited references (e.g., see M.P.E.P. § 2141.02; see also *Stratoflex, Inc., v. Auroquip Corp.*, 713 F.2d 1530, 218 USPQ 871 (Fed. Cir. 1983)). The Examiner must consider each and every element of all of the claims.

Applicant reiterates that independent claim 19 recites, *inter alia*, a method of reducing a lack of performance in a computer system having at least one primary node and a secondary node, including:

> determining whether a failure or lack of performance is imminent;
> based on said determining, commanding a secondary node to start an application if it is not already running, and to begin reading a state and redo log from a memory coupled to said primary node and said secondary node;
> commanding the secondary node to apply the redo log to its state;
> commanding the primary node to begin mirroring its dynamic state updates to the secondary node as they occur, such that the secondary node's state is brought completely up to date with said primary node;
> judging whether the primary node has failed; and
> based on said judging, making the secondary node become the primary node (emphasis added).

Thus, according to the novel and unobvious features of the present invention, a system that is about to fail (e.g., the first node, or the "failing computer", etc.) can "prime" another computer (e.g., the secondary node, or the "failover target", etc.) by directing it to read from disk the stale state and the redo log, and then direct the failover target computer to begin applying the redo log to the stale state (e.g., see specification at page 5, lines 19-23).

Moreover, upon discovery of its imminent demise, the failing computer could mirror all transactions to the failover target, bringing the failover target's state even more up-to-date. When the primary computer does finally fail, the failover target would have an up-to-date copy of the state, the lengthy reloading of state and redo log from disk would be avoided, the outage would be shortened, and the system availability would be improved (e.g., see specification at page 6, lines 7-12).

Thus, unlike Fulton or Garg, the claimed invention reduces outage duration by using the ability to predict outages to proactively trigger and execute functionality whose effect is to reduce that outage's duration. Further, according to the claimed invention, in the case of a single node, the amount of time required to reconstruct the node's state after the outage has occurred can be reduced significantly (e.g., see specification at page 6, lines 7-12).

Applicants submit that neither Fulton nor Garg discloses, suggests, or for that matter, even mentions reducing outage duration for that outage by using the ability to predict outages to proactively trigger and execute functionality whose effect is to reduce that outage's duration, as in the claimed invention.

Instead, Fulton discloses (at best) performing a rejuvenation during a most idle time, which takes less time than performing a rejuvenation during a non-idle time (see Fulton at

column 5, lines 55-60). That is, Fulton does not reduce the outage time for that particular outage, but instead, selectively performs the outage at a time where the outage time is comparatively less than some other outage time.

Indeed, Fulton merely relates to software rejuvenation technique which improves reliability of a software system by periodically stopping execution of the system and restarting the execution with a "clean" state (e.g., see Fulton at Abstract; see also column 2, lines 11-17; emphasis added). That is, Fulton merely discloses the concept of rejuvenating the software.

However, contrary to the Examiner's position, Fulton does not disclose or suggest "*a method for reducing a time for a computer system to recover from a degradation of performance in a hardware or a software*" (e.g., see Office Action at page 2, lines 14-17; *citing* Fulton at column 2, lines 11-23).

Indeed, at column 2, lines 11-23, Fulton does not disclose, suggest, or even mention reducing outage duration or reducing a time for a computer system to recover from a degradation of performance, as in the claimed invention. Instead, Fulton simply states that:

> The invention is based on the observation that the probability that an application will fail increases with the length of time that application continues to run. Consequently, failure can be prevented by periodically stopping execution of the application and restating the execution with clean internal state. That procedure is termed herein rejuvenation. One way of rejuvenating an application is to take advantage of the fact that a newly-created process has clean internal state. Thus, an application can be rejuvenated by terminating the process which is presently executing the application and restarting the application so that it is executed by a new process.

(see Fulton at column 2, lines 11-23; emphasis added).

Accordingly, Fulton merely discloses the concept of rejuvenating the software, not reducing outage duration of that outage or reducing a time for a computer system to recover from

a degradation of performance, as in the claimed invention (nor has the Examiner cited such disclosure in Fulton).

In Fulton, an application is registered with the watchd daemon (i.e., the monitor), and watchd monitors the application. If watchd detects that the process which is currently executing the application is dead or hung, watchd restarts the application on a new process (e.g., see Fulton at column 2, lines 33-35; emphasis added). That is, as the Examiner acknowledges, in Fulton the watchd does not do anything until the application is dead or hung.

Moreover, Fulton does not disclose, suggest, or even mention reducing the actual duration of performing that rejuvenation (e.g., the outage duration).

On the other hand, Garg relates to a method of detecting and estimating aging in operational software. Garg monitors operating system resource usage and system activity and discloses using an "Estimated time to exhaustion" metric to compare the effect of aging on different system resources and also in the identification of important resources to monitor and manage.

In other words, Garg is concerned with identifying the time (i.e., the occasion) for performing the rejuvenation. That is, Garg is trying to predict when to perform the rejuvenation such that the operational software can be stopped and then restarted in a clean state, hopefully before the operational software actually fails (e.g., see Garg at page 1, column 2, first paragraph, lines 7-10).

Similar to Fulton above, Garg also does not, however, disclose, suggest, or even mention reducing the actual duration of performing *that* rejuvenation (e.g., the outage duration), but rather, tries to predict when to perform the rejuvenation process (i.e., at a convenient time). This

has no affect on the duration of *that* rejuvenation process (e.g., the outage duration), and Garg clearly does not disclose, suggest, or even mention that such is the case.

Instead, Garg specifically states that the disclosed approach is "*a first step towards predicting aging related failure occurrences, and may help us in developing a strategy for software fault-tolerance approaches, such as software rejuvenation, triggered by actual measurements*" (e.g., see page 10, first paragraph; emphasis added).

For the foregoing reasons, Applicants submit that the methods of both Fulton and Garg are entirely unrelated to limiting the duration of a particular rejuvenation process (i.e., that rejuvenation process, as opposed to one rejuvenation process in comparison with another rejuvenation process), or for that matter, limiting the duration for failure recovery processes, as in the claimed invention.

Accordingly, Applicants respectfully submit that it would not have been obvious to combine Fulton and Garg to arrive at the claimed invention as defined by claim 19 because neither Fulton nor Garg, when considered as a whole for what they fairly teach to the ordinarily skilled artisan) even contemplates the problems addressed and solved by the claimed invention, when considered as a whole.

Moreover, Applicants submit that, even assuming *arguendo* that it would have been obvious to modify Fulton in view of Garg, the resulting combination clearly would not disclose or suggest the claimed invention, which reduces a particular outage duration by using the ability to predict outages to proactively trigger and execute functionality whose effect is to reduce that outage's duration, as in the claimed invention.

Instead, the combination would result (at best) in a rejuvenation process as described in Fulton being performed on the occasion described by Garg, not a method that reduces the actual

time for the computer system to recover (e.g., the outage duration) from a degradation of performance (or a rejuvenation process), as in the claimed invention.

For the foregoing reasons, Applicants respectfully submit that Fulton and Garg, either alone or in combination, do not disclose or suggest all of the novel and unobvious features of independent claim 19.

Independent claims 26-28 and 34-40

Applicants submit that independent claims 26-28 and 34-40 recite somewhat similar features as independent claims 1 and 19, and therefore, are patentable over Fulton and Garg for somewhat similar reasons as those set forth above.

For example, independent claim 26 recites a method of maintaining performance of a computer system, comprising:

> monitoring a primary node of said computer system;
> determining whether the primary node is failing or about to fail; and
> migrating the state of the primary node to another node in said computer system,
> wherein there is other than a one-to-one relationship between the another node and the primary node (emphasis added).

Independent claim 27 recites a method of reducing a degradation period of a Web hosting machine, comprising:

> monitoring a performance of said Web hosting machine; and
> transferring a state of said Web hosting machine to a second machine when a degradation of said performance occurs in said Web hosting machine (emphasis added).

Further, independent claim 28 recites a method of reducing a degradation of performance in a computer system having at least one primary node and a secondary node, comprising:

> determining whether a degradation of performance of said primary node is imminent;
>
> based on said determining, commanding said secondary node to start an application if it is not already running;
>
> replicating, by said secondary node, a state of said primary node; and
>
> passing control to said secondary node from said primary node (emphasis added).

On the other hand, independent claim 34 recites a method of reducing a degradation of performance in a computer system having a single node, comprising:

> determining whether a degradation of performance of the node is imminent; and
>
> based on the determining, commanding the node to begin storing its state on a stable storage at a more frequent rate, to reduce a staleness of the state on the stable storage (emphasis added).

Further, independent claim 35 recites a system for reducing a time for a computer system to recover from a degradation of performance in a hardware or a software in at least one first node of said computer system, comprising:

> a monitor for monitoring a state of said at least one first node; and
>
> a transfer mechanism for, based on an output from said monitor, transferring a state of said at least one first node to a second node prior to said degradation in performance of said hardware or said software of said at least one first node (emphasis added).

Independent claim 36 recites a computer system, comprising:

> at least one first node;
>
> a second node;
>
> a shared memory coupled to said first and second nodes;
>
> a monitor for monitoring a state of said at least one first node; and
>
> a transfer mechanism for, based on an output from said monitor, transferring a state of said at least one first node to said second node prior to a degradation in performance of hardware or software of said at least one first node (emphasis added).

Independent claim 37 recites a system for reducing a degradation of performance in a computer system having a single node and a stable storage, including:

> a monitoring unit for monitoring whether a degradation of performance of the node is imminent; and
> a transfer mechanism for, based on an output from the determining unit, commanding the node to begin storing its state on a stable storage at a more frequent rate, to reduce a staleness of the state on the stable storage (emphasis added).

Independent claim 38 recites a signal bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of reducing a time for a computer system to recover from a degradation of performance in a hardware or a software in at least one first node of said computer system, the method comprising:

> monitoring a state of said at least one first node; and
> based on said monitoring, transferring a state of said at least one first node to a second node prior to said degradation in performance of said hardware or said software of said at least one first node (emphasis added).

On the other hand, independent claim 39 recites a signal bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of reducing a degradation of performance in a computer system having a single node, the method comprising:

> determining whether a degradation of performance of the node is imminent; and
> based on the determining, commanding the node to begin storing its state on a stable storage at a more frequent rate, to reduce a staleness of the state on the stable storage (emphasis added).

Independent claim 40 recites a method of reducing a time for a computer system to recover from a degradation of performance in a hardware or a software in a node of the computer system, comprising:

> monitoring a state of said node; and
> based on said monitoring, transferring a state of said node to one of a stable storage and another node prior to said degradation in performance of said hardware or said software of said node (emphasis added).

Applicants respectfully submit that Fulton and Garg, either alone or in combination, do not disclose or suggest all of the novel and unobvious features of independent claims 26-28 and 34-40.

For example, Fulton and Garg, either individually or in combination, do not disclose or suggest "*determining whether the primary node is failing or about to fail... and migrating the state of the primary node to another node*", as defined by claim 26.

As another example, Fulton and Garg, either individually or in combination, do not disclose or suggest "*determining whether a degradation of performance of the node is imminent; and based on the determining, commanding the node to begin storing its state on a stable storage at a more frequent rate, to reduce a staleness of the state on the stable storage*" (emphasis added), as defined by claims 34 and 37-39.

Applicant submits that dependent claims 2-18, 20-25, and 29-33 also are patentable over the cited references by virtue of their respective dependencies from independent claims 1, 19, and 28, as well as for the additional features recited therein.

Accordingly, the Examiner respectfully is requested to withdraw the rejection of claims 1-3 and 5-40 and permit these claims to pass to immediate allowance.

B. Claim 27 stands rejected under 35 U.S.C. § 103(a) as being obvious over Fulton in view of Garg, and further in view of Murphy.

For somewhat similar reasons as those set forth above, Applicants respectfully reiterate that it clearly would not have been obvious to combine Fulton and Garg to arrive at the claimed invention defined by claim 27, since neither Fulton nor Garg discloses, suggests, or even mentions "reducing a degradation period" as claimed.

For example, independent claim 27 recites, *inter alia*, a method of reducing a degradation period of a Web hosting machine, including:

> monitoring a performance of said Web hosting machine; and
> transferring a state of said Web hosting machine to a second machine when a degradation of said performance occurs in said Web hosting machine (emphasis added).

Moreover, Applicant reiterates that Murphy would not have made up for the deficiencies of Fulton and Garg. Indeed, Murphy is not even relied upon for the features of reducing a degradation period or reducing a time to recover from a degradation of performance in a hardware or a software, as claimed. Instead, Murphy is relied upon for showing "*a node for a Web hosting machine*" (see Office Action at page 23, lines 7-8).

Thus, Applicants submit that Fulton, Garg, and Murphy, either alone or in combination, do not disclose or suggest all of the novel and unobvious features of the claimed invention.

Therefore, the Examiner is requested to withdraw this rejection and permit claim 27 to pass to immediate allowance.

III. CONCLUSION

In view of the foregoing, Applicants submit that claims 1-3 and 5-40, all the claims presently pending in the application, are patentably distinct over the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Assignee's Deposit Account No. 50-0510.

Respectfully Submitted,

Date: February 14, 2005

John J. Dresch, Esq.
Registration No. 46,672
Sean M. McGinn, Esq.
Registration No. 34,386

McGinn & Gibb, PLLC
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 21254

CERTIFICATE OF TRANSMISSION

I certify that I transmitted via facsimile to (703) 872-9306 the enclosed after-final Amendment under 37 C.F.R. § 1.116 to Examiner Joshua A. Lohn on February 14, 2005.

John J. Dresch, Esq.
Registration No. 46,672
Sean M. McGinn, Esq.
Registration No. 34,386